MAI_simulator
Micro Array Image Simulator
USERS MANUAL

Version 0.5 / July 2007 Edition

Nicolas Mary & Christophe Klopp

Revision table :

Version	*Date*	*Major modifications*	*Author(s)*
0.1	June 6th 2007	First draft	N. Mary
0.2	June 27th 2007	First revision	Ch. Klopp
0.3	June 27th 2007	Second revision	N. Mary
0.4	July 11th 2007	Third revision	Ch. Klopp
0.5	July 11th 2007	Fourth revision	N. Mary

Acknowledgment :
I wrote this guide in English having the aim to give access to MAI_simulator to the largest possible scientific community. But English is not my mother language, so any remark about the orthography or the grammar is welcome.

INDEX

Illustration Index

I. Introduction

Micro Array Image Simulator (MAI_simulator) has been developed in order to generate micro array images. These images are simulated thanks to several user defined parameters. Generating these images aims at creating test sets for image quantification software. For each run the simulator generates sets of images and the corresponding quantification files. These quantification files can be used as references while analyzing and comparing microarray image quantification software.

Parameters are sorted in five categories :

- General parameters
- Grid parameters
- Spot parameters
- Noise parameters
- Spike-ins

General parameters define the naming schema of the created files and the directory in which the file are generated. The grid parameters define the grid configuration elements such as number of meta rows, meta columns, rows and columns. The spot parameters determine the spot shape, expression level , saturations and deformation. In order to stick to real data users are able to add different kinds of noises thanks to noise parameters. The spike-ins tab enables the user to define the location of saturated spots in the image. The spots can be used in the grid positioning process or in quantification normalization process.

When all parameters fields are set, the user can launch the process and then take a look at the generated images and quantification files.

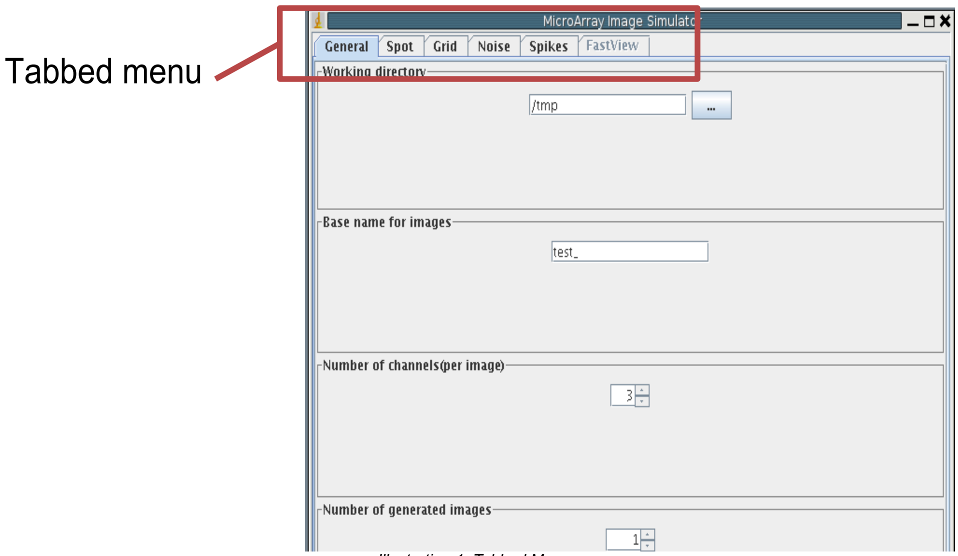

Illustration 1: Tabbed Menu

The launch button "Validate" and the progress bar are located at the bottom of the pane.

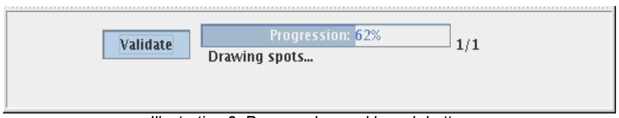

Illustration 2: Progress bar and launch button

II. Parameter Setting

The chapter explains how to set the different parameter fields before launching the process.

1. General

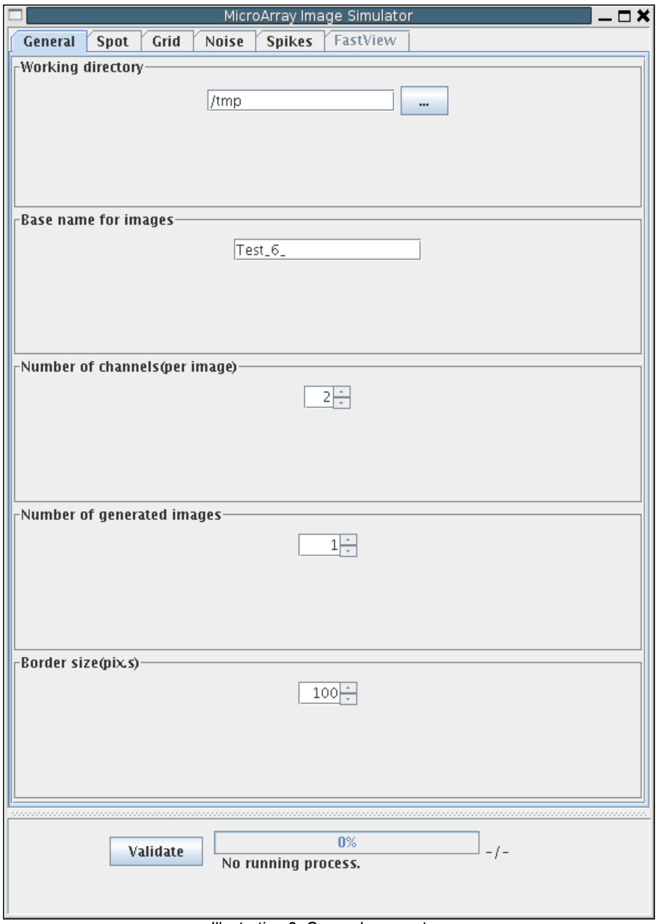

Illustration 3: General parameters

The general tab enables to specify :

- **Working directory**
  This is the directory where all output files will be saved.

- **Number of channels**
  The number of channels of your experiments (between 1 and **)

- **Number of images**
  The number of experiments you want to generate (between 1 and **)

- **Base name**
  All images and quantification files names will start with the base name. For example, if you choose a 3-channels, 2 images experiment with "Test" as base name, output files will be :
  -Test1_1.tiff, Test1_2.tiff, Test1_3.tiff (the 3 channels of the first image)

-Test2_1.tiff, Test2_2.tiff, Test2_3.tiff (the 3 channels of the second image)
-Test1.txt (the quantification file for the first image)
-Test2.txt (the quantification file for the second image)

- **Border size (in pixels)**
  You can set the size of the image borders.

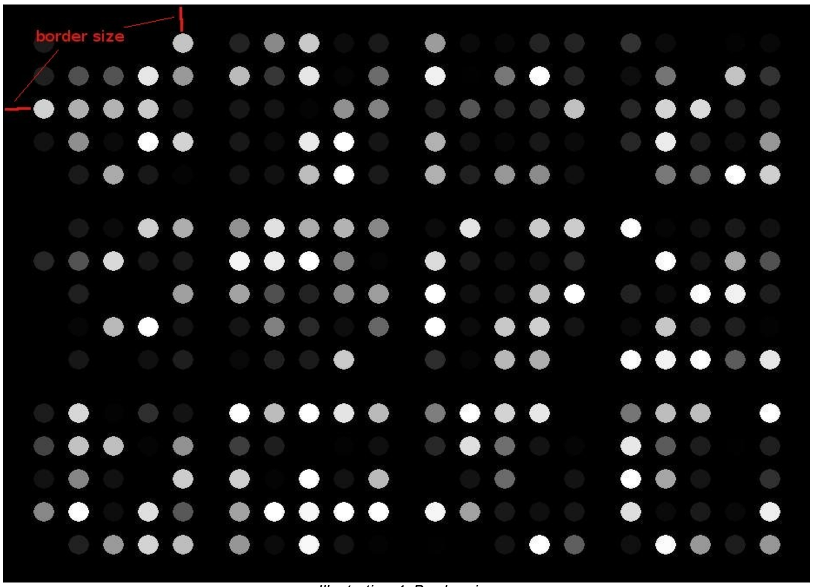

Illustration 4: Border size

NB : Be careful when defining the border size for rotated images.

2. Grid

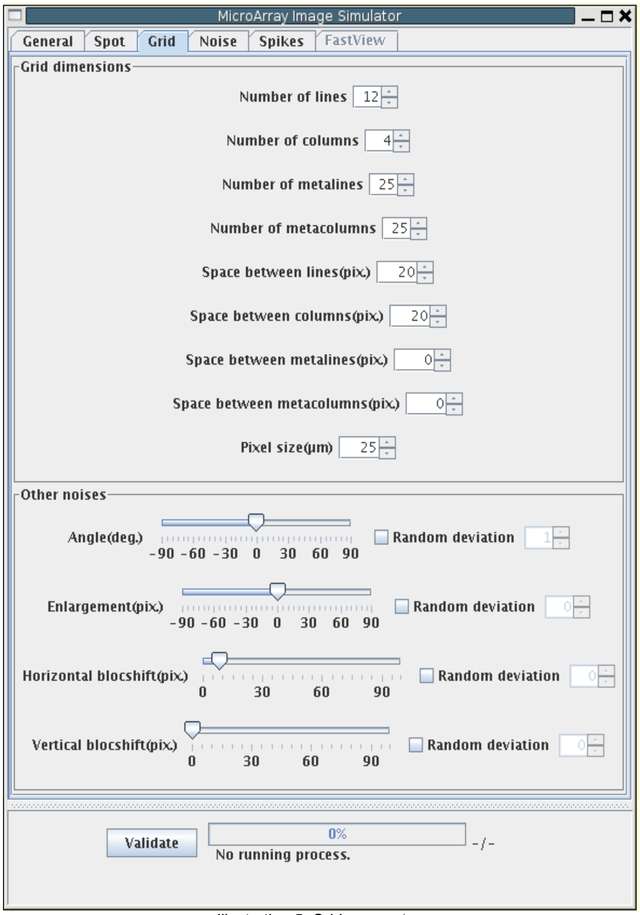

Illustration 5: Grid parameters

1. Grid dimensions

- **Number of lines**
  This is the number of block lines.
- **Number of columns**
  This is the number of block columns.
- **Number of meta-lines**
  This is the number of lines in one block.
- **Number of meta-columns**
  This is the number of columns in one block.
- **Space inter-lines (in pixels)**
  This is the gap between block lines.
- **Space inter-columns (in pixels)**
  This is the gap between block columns.
- **Space inter-meta-lines (in pixels)**
  This is the gap between lines in each block.
- **Space inter-meta-columns (in pixels)**
  This is the gap between columns in each block.
- **Pixel size (in µ-meters)**
  This is the pixel height and width.

2. Grid noises

For not perfectly regular grids it is possible to introduce three kinds of noise :

- **Angle (in degrees)**
  Rotates the grid.
- **Enlargement (in pixels)**
  Enlarge the grid length.
- **Block shift (in pixels)**
  Make a random shift on the grid blocks. You must give two values for each shift side (horizontal and vertical) : the mean and the standard deviation. More explanations are given in the “*Noises and random”* chapter.

3. Spot

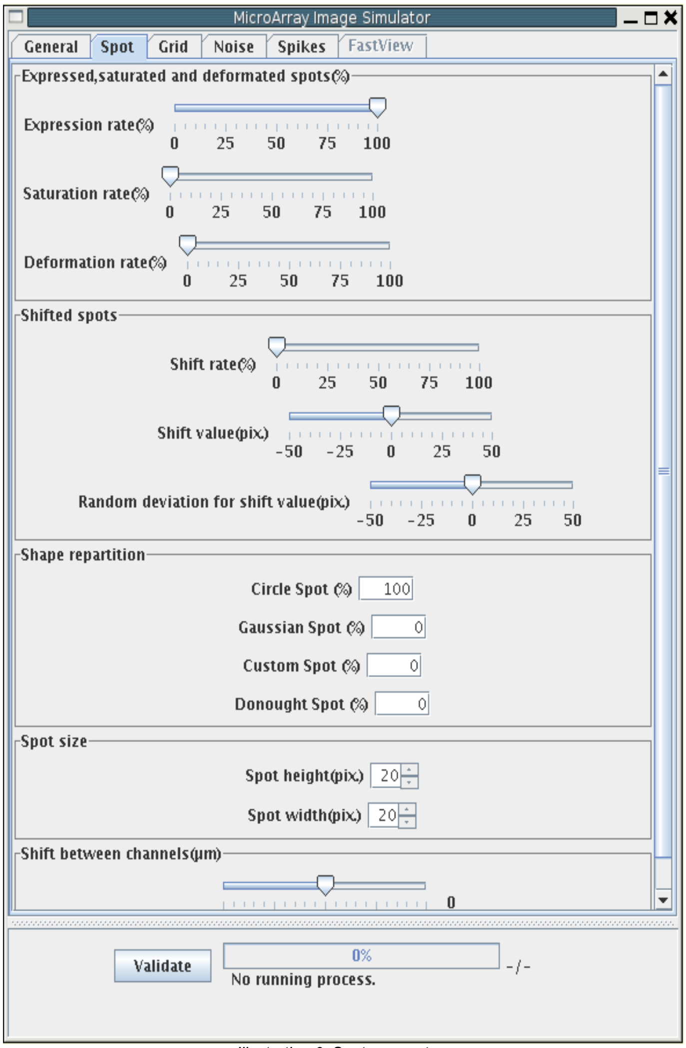

Illustration 6: Spot parameters

1. General

Once grid parameters have been set, you have to decide what you want the spots to look like.

- **Expression rate (%)**
  Sets the rate of expressed spots.
- **Saturation rate (%)**
  Sets the rate of saturated spots.
- **Deformation rate (%)**
  Sets the rate of distorted spots.
- **Shape repartition (%)**
  Here, you will choose a percentage for each shape model. The total must be 100 unless the process can't be launched.
- **Spot size (in pixels)**
  You must also set the spot width and height.

2. Spot noises

- **Spot shift (in pixels)**
  Make a random shift on spots. You must give two values : the mean and the standard deviation. More explanations are given in the "*Noises and random"* chapter.
- **Inter-channel shift (in pixels)**
  In mono-channel case, this has no effect. In multi-channel case, this noise is not applied to the first channel image but to all others.
  Make a random shift on the whole channel image. You just give one value here : the standard deviation. In fact, the mean is set to 0. More explanations are given in the "*Noises and random"* chapter.

4. Noise

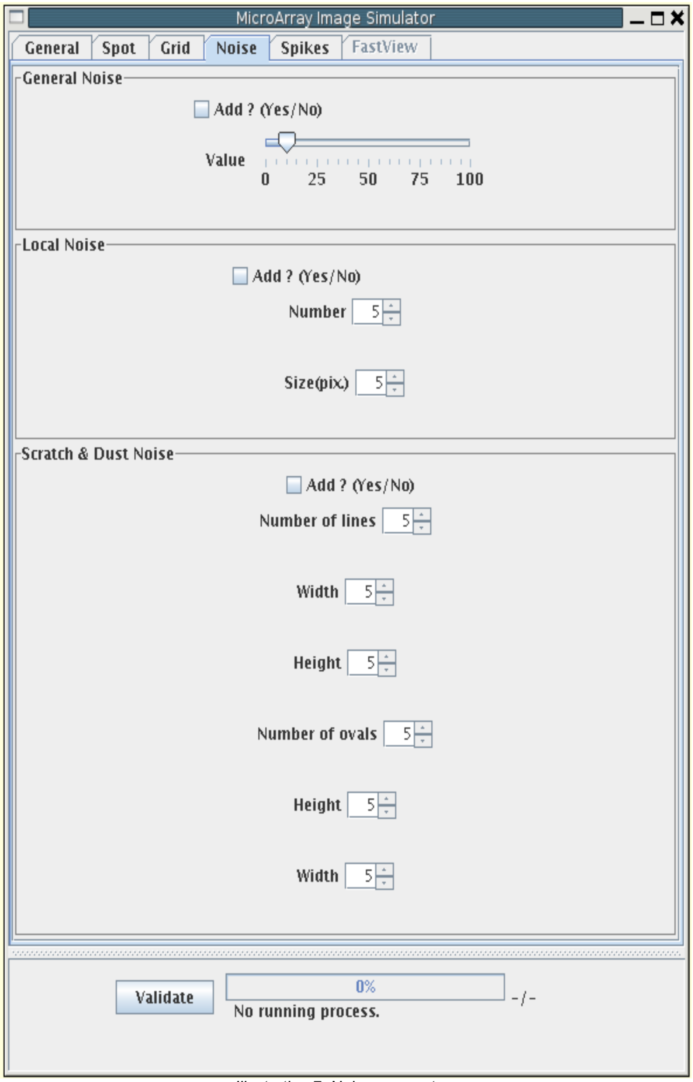

Illustration 7: Noise parameters

1. General noise

Here you can set a background noise value. This will add a “salt and pepper” noise all over the images. The value sets the noise density : a high value means a high density.

2. Local noise (*Not implemented in the July 2007 version*)

Local noise refers to fractal phenomenas. If you want to add this kind of noise, you must choose the number of fractals and their size. During the process, fractals will be placed randomly on the images.

3. Scratch & dust noise

You can also put scratch and dust noises on your images. These aim to simulate the micro-dusts on the glass slides. You could choose two shapes : ovals and/or lines, and how much of each you want. Then you must customize them specifying their width and height. Like with local noise, scratches and dusts will be placed randomly on the images.

5. Spikes

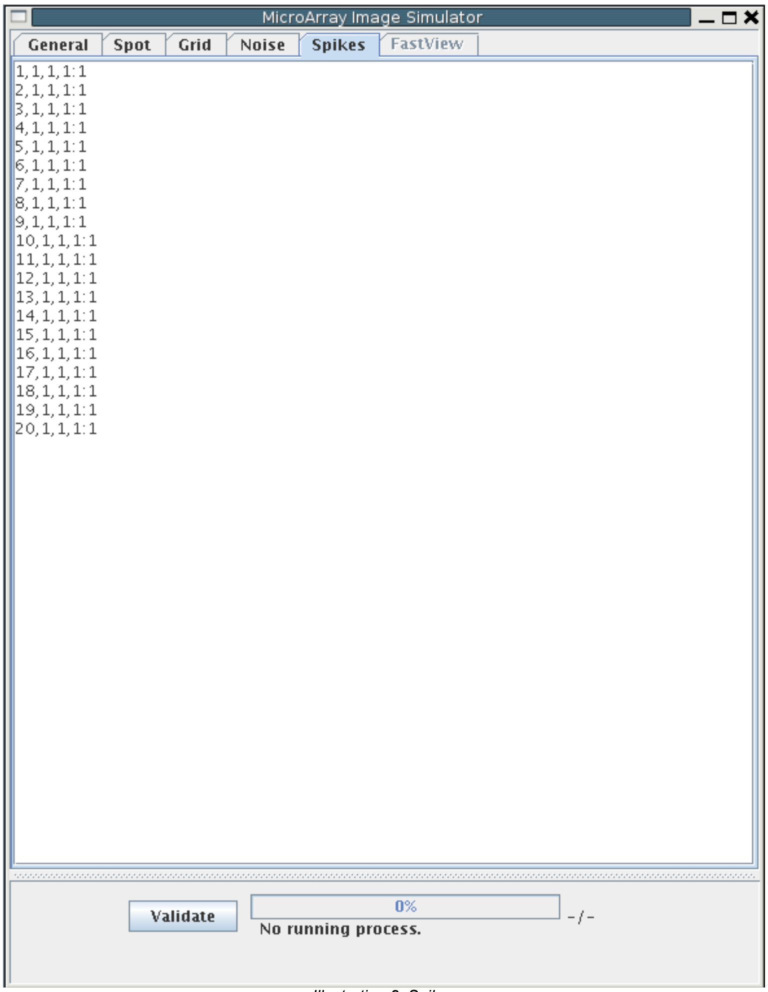

Illustration 8: Spikes

Spikes are spots set to the maximum intensity value. In fact, they are used to indicate a position on the grid, like tags. They are very helpful in the alignment process.
For example, they can be used to locate the upper-left corner of each block of the grid.
You can specify a list of spikes giving their coordinates with the following format :

i,j,k,l:c1{,c2,c3,...} *{...}* means *optional*

i is the line index of the block
j is the column index of the block
k is the line index of the spot in the block
l is the column index of the spot in the block
c1,c2,c3 are the channel indexes, you must give one at least

6. Fastview

When MAI_Simulator has ended producing the tif files (16bits , one file per channel), the user can visualize the upper left corner of the generated images using the fast view panel. He can choose the image he wants to check in the drop-down list.

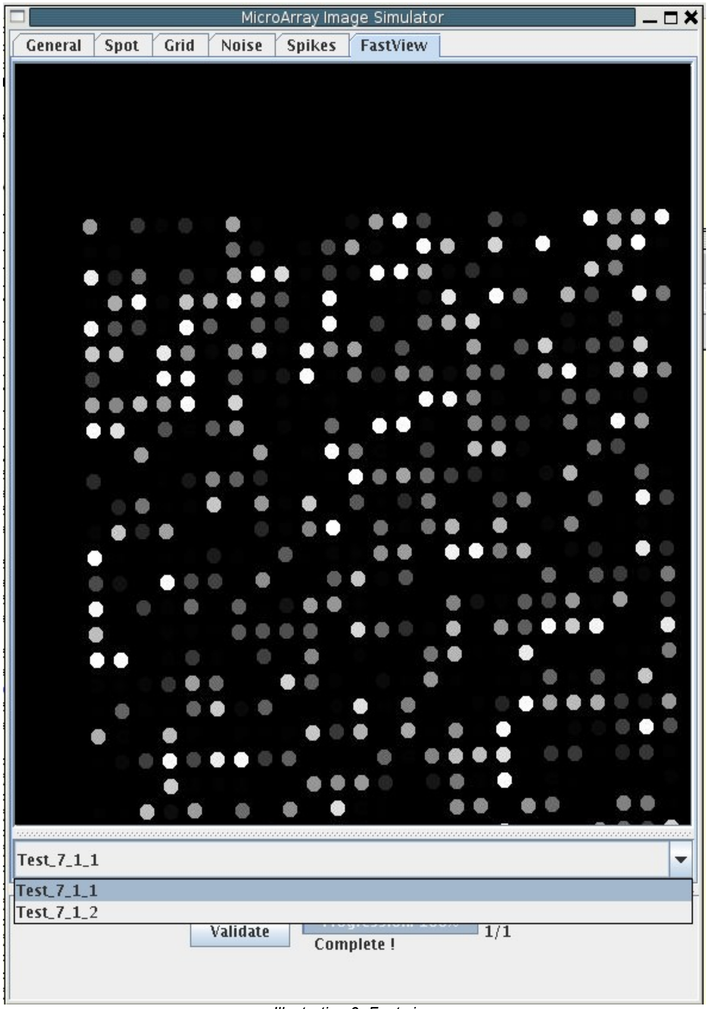

Illustration 9: Fast view

7. Result file

MAI_Simulator generates images on one side and quantification files on the other. The quantification files contain the random channel intensities for each channel of the image. The examples presented hereunder correspond to single and a four channel images.

These files have several columns :

- Index : spot index
- X and Y location in micro meters of the spot center on image,
- R, C, MR, MC : bloc row, bloc column, spot row and spot column within the bloc,
- ChanX : X channel intensity

The result file is used to evaluate the quantification accuracy of microarray image analysis tools.

"Creator=ImageJ - MAIS Plugin"
"File=1_All_Normal1.txt"
"PixelSize=25"
"ImageOrigin=0,0"

Index	X	Y	R	C	MR	MC	Chan1
1	8925	5525	1	1	1	1	27556
2	9425	5550	1	1	1	2	3721
3	9925	5600	1	1	1	3	56169
4	10425	5650	1	1	1	4	144
5	10925	5700	1	1	1	5	45369
6	11400	5725	1	1	1	6	44521
7	11900	5775	1	1	1	7	32400
8	12400	5825	1	1	1	8	85756
9	12900	5875	1	1	1	9	169
10	13400	5900	1	1	1	10	225
11	13900	5950	1	1	1	11	676
12	14425	6000	1	1	1	12	361
13	8875	6025	1	1	2	1	1521
14	9375	6050	1	1	2	2	43681
15	9875	6100	1	1	2	3	676
16	10375	6150	1	1	2	4	225
17	10875	6200	1	1	2	5	256
18	11375	6225	1	1	2	6	58081
19	11875	6275	1	1	2	7	169
20	12375	6325	1	1	2	8	15625

Illustration 10: Quantification file example 1

"Creator=ImageJ - MAIS Plugin"
"File=4_Sinus_Normal3.txt"
"PixelSize=25"
"ImageOrigin=0,0"

Index	X	Y	R	C	MR	MC	Chan1	Chan2	Chan3	Chan4
1	8600	5950	1	1	1	1	49	92416	30976	81
2	9100	5975	1	1	1	2	61009	61009	61009	61009
3	9575	6000	1	1	1	3	44100	44100	44100	44100
4	10075	6050	1	1	1	4	20164	20164	20164	20164
5	10575	6075	1	1	1	5	6561	6561	6561	6561
6	11000	6100	1	1	1	6	2116	2116	2116	2116
7	11575	6150	1	1	1	7	26569	26569	26569	26569
8	12100	6200	1	1	1	8	100	100	100	100
9	12600	6225	1	1	1	9	441	441	441	441
10	13100	6250	1	1	1	10	1849	57600	93636	40401
11	13600	6300	1	1	1	11	49	49	49	49
12	14100	6325	1	1	1	12	3721	3721	3721	3721
13	8550	6475	1	1	2	1	61009	61009	61009	61009
14	9050	6475	1	1	2	2	484	484	484	484
15	9550	6500	1	1	2	3	441	441	441	441
16	10050	6500	1	1	2	4	3136	3136	3136	3136
17	10550	6575	1	1	2	5	64	64	64	64
18	11050	6625	1	1	2	6	121	121	121	121
19	11550	6650	1	1	2	7	16	16	16	16
20	12075	6675	1	1	2	8	5929	5929	5929	5929
21	12575	6725	1	1	2	9	70225	70225	70225	70225
22	13075	6750	1	1	2	10	16384	16384	16384	16384
23	13575	6800	1	1	2	11	30625	30625	30625	30625
24	14075	6825	1	1	2	12	289	289	289	289

Illustration 11: Quantification file example 2

III. Noises and random values

Random number generation is very useful to produce a realistic set of images. Nearly all kinds of noise have a random part. This chapter explains how random is used in MAI_Simulator.

DNA chips images are known to contain many kinds of noise :

Noise in experiment chain	**Noise in MAISimulator**
Systematic position errors because of pin deformations.	Block shift noise
Non systematic position error	Spot shift noise
Orientation variation	Angle noise
Deformation of the initial spotting schema due to soft support material (membrane technology)	Enlargement noise
Background noise due to quality of scan	General and local noises
Background noise due to dusts	Scratch and dust noise

1. Mean value and standard deviation

Some random values are generated using the Gaussian distribution. This is why, often you have to give two values in the user interface of MAI_Simulator : the mean value and the standard deviation.

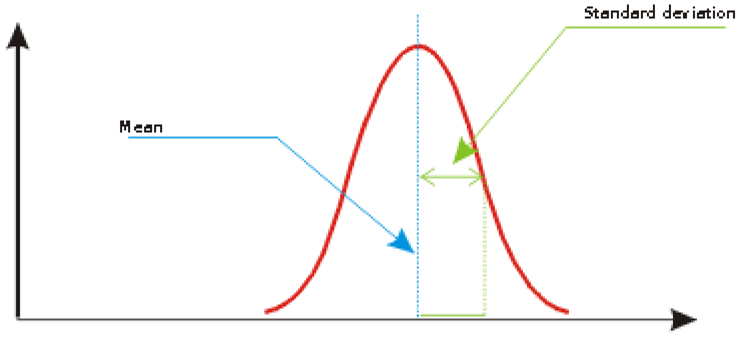

Illustration 12: Gaussian curve

The values are taken between (**Mean** - 3***Sd**) *and* (**Mean** + 3***Sd**).
For example, if you give a mean value of 2.0 and a standard deviation of 1.0, the random values could be between -1.0 *and* 5.0 but in most of cases they will be between 1.0 *and* 3.0.

2. Spot repartition

The spot positioning in image uses a random system too. In fact, you give the system the percentage you want for each spot shape, then according to the total number of spots and these percentages the software computes how much spots of each shape are needed. For each spot, it chooses the shape at random and re-samples if the quantity needed has been

reached. Finally, you obtain exactly the percentages you have specified but the space repartition over images is randomly designed.

Adding spot shapes

Three shape models have been developed in this Plugin : Circle, Gaussian and Donut. MAI_Simulator was designed in order to allow developers to easily add new ones without changing the code.

Each shape model implements an interface class : the Spot class. Examples can be found using the CircleSpot, GaussianSpot and DonutSpot classes.

All the class names end with "Spot" and this is used by MAI_Simulator to load them. So if someone wants to develop a new shape model, she or he has to create a new class named "*ShapeName*Spot" derived from Spot. After that two abstract methods ***getImageSpot*** and ***deformate*** must be implemented.

Then the **java file** must be compiled and the **class file** put in the "/**plugins" directory** of your ImageJ install.

Finally, the **MAIS_parameters.properties** file must be updated by adding a new spot shape name to the shapes line :
shapes=Circle,Gaussian,Donut,*ShapeName*

*where **ShapeName** is the new shape model.*

If a model is to be removed just remove its name from the list.
Example removing the Circle shape :
shapes=Gaussian,Donut,*ShapeName*

The user interface evolves with the shape names provided in the shapes line of the **MAIS_parameters.properties** file.

User interface before :

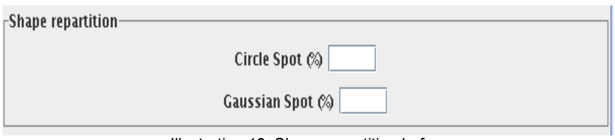

Illustration 13: Shape repartition before

User interface after adding Custom and Donought shapes:

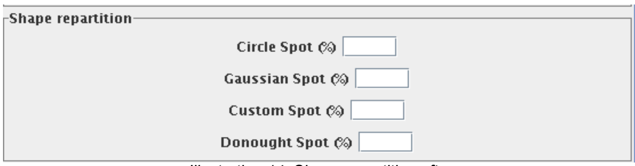

Illustration 14: Shape repartition after

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA  02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.